Multi-Robot Task Allocation in Lunar Mission Construction Scenarios

George Thomas
University of Iowa
Department of Computer Science,
Iowa City, IA, USA
george-thomas@uiowa.edu

Ayanna M. Howard
Georgia Institute of Technology
School of Electrical and Computer Engineering,
Atlanta, GA, USA
ayanna.howard@ece.gatech.edu

Andrew B. Williams
Spelman College
Department of Computer and Information Sciences
Atlanta, GA USA
williams@spelman.edu

Aryen Moore-Alston
Spelman College
Department of Computer and Information Sciences
Atlanta, GA USA
amoore-a@spelman.edu

Abstract.– *In this paper, we propose a method for multi-robot task allocation based on the concept of task decomposition for a lunar mission scenario. This methodology focuses on segmenting a task scenario into a sequence of operations called functional primitives that are defined a priori by a set of performance metrics and resource requirements. In real-time, multiple robotic agents determine their capabilities and skill sets associated with the defined functional primitives in order to determine a suitable allocation scheme. We discuss the methodology in detail, and provide results for a simulated lunar mission construction scenario using the Multi-Agent Robot Simulator for Lunar Construction (MARS-LC) system.*

Keywords: multi-agent coordination, task allocation, multi-robot systems, space exploration.

1 Introduction

The problem of how to intelligently coordinate robots to work together in teams has a wide application in robotics and multi-agent systems. As robots continue to become integrated into our society, it is envisioned that the application of robot teams will see increased usage as they work autonomously without any human intervention. In scenarios of teams mixed with humans and robots, human participants may need to offload tasks to the robots. Then the robots with little or no human supervision will need to readily assume this workload while effectively allocating the tasks among themselves, monitoring each other, and performing task reallocation or reorganization as necessary.

One of the primary applications for robot teams is in the area of space exploration. A new vision for space exploration was established in [1], which promotes the development of new approaches for sustainable human and robotic technologies for lunar missions. As part of this vision, multi-agent teaming using adjustable autonomy human-robot teams was presented as one of the key technologies to enable these future missions.

In this paper, we address the issue of multi-agent teaming for lunar construction using our Multi-Agent Robot Simulator for Lunar Construction (MARS-LC) simulation environment. The paper is organized as follows. In section 2 we describe related work. Section 3 details the problem of lunar construction using robot teams. In section 4, we describe our mechanism for task allocation in robot teams engaged in autonomous construction of lunar habitats. In section 5 we detail how we evaluated our approach and we present our conclusions in section 6.

2 Related Work

Research in multi-agent task allocation [2, 3] has focused on decentralized methods for allocating tasks among autonomous agents in uncertain and dynamic environments by maximizing an optimization function that correlates to overall system performance. The optimization function is calculated based on various techniques, ranging from using reward functions in Markov decision processes [4] to defining utilities in optimal assignment problems [3].

[5] was one of the first approaches that used market-based systems to allocate tasks. A planner was incorporated to allow robots to assess the costs of each task. The MURDOCH system [6] uses an auction-based system to allocate tasks in an online manner. Tasks were allocated to the auction winner by means of a time-limited contract which was periodically renewed if adequate progress was being made in the task achievement. [7] used role switching to investigate task allocation in a simulated system of robots working on cooperative transport tasks. The basic algorithm in all these methods is a greedy strategy that is not guaranteed to yield an optimal allocation [3]. [8] and [9] examined behavior based approaches to the task allocation problem that also use a greedy algorithm. Greedy strategies require some means of assessing cost, utility, reward, eligibility or capability as the appropriate method calls this metric. Hence, the precise definition of such a metric greatly influences the performance of any greedy strategy. [10] examined the problem of complex task

allocation whereby a task initially assigned to a single robot can be further broken down into subtasks by that robot and auctioned off to others. Robots are allowed to bid on any node of a task decomposition tree instead of just the root node. The costs are controlled by each robot's specific, possibly different, plan for implementing that node.

Our unique contribution is to apply task allocation strategies to an applied problem involving a variety of different tasks that have constraints between them. General strategies have been designed for efficient task allocation but further improvement depends on knowledge of the problem domain. Knowledge of the world model may give insight into how tasks are generated and will be required if learning strategies are used. We use a market-based algorithm that can be applied to the different stages of tasks in our selected problem. This algorithm handles complex, decomposable tasks and their resulting subtasks. We use an abstract simulation of a multi-robot scenario to illustrate how task allocation can be applied to the complex problem of achieving effective cooperation between robots. We assume a high level hierarchical plan for the task scenario that we are examining. This plan can be provided by a human operator or generated by a planner. While we tailor our approach to our selected problem, we also believe that our strategies can be applied to similar real world problems.

3 Problem Specification

Given a task scenario where there are a set of inter-related domain tasks to be accomplished, and a set of robots among which to allocate these tasks, we seek to address the issue of task allocation in such a way so as to complete the task list in minimal time with minimal consumption of resources. These conflicting requirements pose a unique problem for our application space in that, if we are minimizing power, we will tend to not use robots in order to conserve their power. Yet, this will correspondingly increase the completion time for the scenario. In this paper, we primarily focus on minimizing time.

Our system is composed of heterogeneous, cooperative robots that possess different capabilities. For our application space, we assume that each individual task can be performed by a single robot. That is, no single task requires the participation of two or more robots for its successful execution. We also assume that the domain world is non-deterministic with error-free communication, and that there may be constraints among the tasks. Namely, some tasks have to be executed in sequence and others can be executed in parallel. Finally, our allocation has to be iterative since the full task list is not known on startup and is gradually revealed as current tasks are executed. In general, the focus of our approach is collaborative. We seek to optimize the use of the available robots to execute the given tasks as efficiently as possible.

To test our methodology, the MARS-LC task simulator is developed to simulate a set of robots with multiple capabilities working on constructing lunar habitats. The construction involves performing a set of tasks subject to sequential constraints. Our goal is to abstractly model the lunar construction problem so that our task allocation problem can be shown to work with the different scenarios that this problem presents. We chose the lunar simulation problem because it allows us to examine a variety of different tasks and scenarios such as iterated assignment of tasks, online assignment of tasks, sequentially constrained tasks, and complex tasks.

3.1 Lunar Construction

NASA has been exploring the possibility of constructing lunar habitats to achieve a sustainable presence on the moon [1]. Robots would play a significant role in such a construction, either by assisting humans or by autonomously performing the construction by working as multi-robot teams. The process required for lunar construction presents challenges not encountered here on earth. Though lunar structures require only 1/6th the load bearing capacity of identical structures on earth, this very lack of gravity, and the presence of vacuum-like conditions render normal construction and excavation tools highly power intensive. In addition, the fine lunar dust or regolith can adversely affect the mechanical parts and mechanisms of any equipment used.

There are two possible approaches to lunar construction that we examined based on [11]. In the inflatable method, a double skin membrane is filled with structural foam. First, the ground is shaped, the un-inflated structure is secured upon it, and then injected with structural foam. The internal compartment is pressurized, and the bottoms of the inflated structure are filled with compacted soil. In the erectable method, various geometrically configured 3D trussed octet or space frame elements are used as the modular building blocks. The shapes can be tetrahedral, hexahedral or octahedral. In this paper, we select the inflatable method of construction to apply our methodology for multi-agent teaming.

The inflatable method for lunar construction can be decomposed into the following steps:

1. Survey and choose location for site.
2. Clear the selected site.
3. Transport materials and equipment to site.
4. Excavate and compact the site to ensure a level area.
5. Place materials for construction in the correct positions and attach them together.
6. Inflate the structure with structural foam and pressurize the internal compartment.

4 Methodology

4.1 Action Definition

Our overall approach is based on the strategy of [12, 13]. Briefly, we break up our habitat construction scenario into a sequence of complex operations, which can further be decomposed into simple operations. Each simple operation is broken up into one or more identical tasks. A single task is made up of a set of functional primitives executed in sequence. We define performance metrics for the tasks and specify parameters and resources required for executing the functional primitives.

Our discussion is based on [14, 15, 16]. We form a set of basic actions that a robot must provide. Following the terminology of [12, 13], we call these basic actions functional primitives. Each robot has a set of capabilities or skill sets that allow it to satisfactorily execute a primitive. A functional primitive may also have parameters such as mass or distance associated with its definition and an associated time and energy value that represents the resources required to perform an atomic unit of that action, based on default values of parameters, if any. Finally, in a non-deterministic world, actions may fail, so an estimated failure rate, based on [17], is also part of the definition of a functional primitive. The failure rates serve to introduce some noise into the system, but cannot incapacitate a robot or any of its capabilities. Robot failure is more closely simulated by energy loss.

Table 1 : Functional Primitives

Functional Primitives	**Capabilities**	**Time/ unit action**	**Energy**	**Parameters**
Traverse (*t*)	Stereo Camera (S), Mobile (M)	Varies, t_t	Varies, e_t	distance, mass
Grasp (*g*)	S, Gripper (G)	Fixed, t_g	Fixed, e_g	
Release (*r*)	S, G	Fixed, t_r	Fixed, e_r	
Lift (*l*)	S, Manipulator Arm (A), G	Fixed, t_l	Varies, e_l	mass
Unload (*u*)	S, A, G	Fixed, t_u	Fixed, e_u	
Excavate (*x*)	Large Manipulator Arm (L), M, S	Fixed, t_x	Varies, e_x	volume
Attach/Mate (*a*)	S, G	Fixed, t_a	Fixed, e_a	
Identify (*i*)	S	Fixed, t_i	Fixed, e_i	
Localize (*z*)	S, Navigation (N)	Fixed, t_z	Fixed, e_z	
Track (*k*)	S, N	Fixed, t_k	Fixed, e_k	
Model (*m*)	Cognitive (C)	Fixed, t_m	Fixed, e_m	
Plan (*p*)	Cognitive (C)	Fixed, t_p	Fixed, e_p	
Recharge (*c*)	S, M, N	Fixed, t_c	Fixed, e_c	

The functional primitives chosen for our simulation are an expanded set of the primitives described in [18].

4.2 Task Definition and Decomposition

The process of constructing an inflatables-based lunar habitat structure can be defined by a set of five primary operations. We iteratively subdivide each operation into simpler tasks until all leaf tasks are at the granularity of functional primitives. The set of primitives grouped together at the leaf node are considered basic actions performed in sequence that constitute a unit task. Any primitive that fails during implementation has to be fully repeated except in cases where energy wasted is a function of distance traveled (such as for robotic traverses). The habitat construction scenario decomposition scheme is represented in Figure 1.

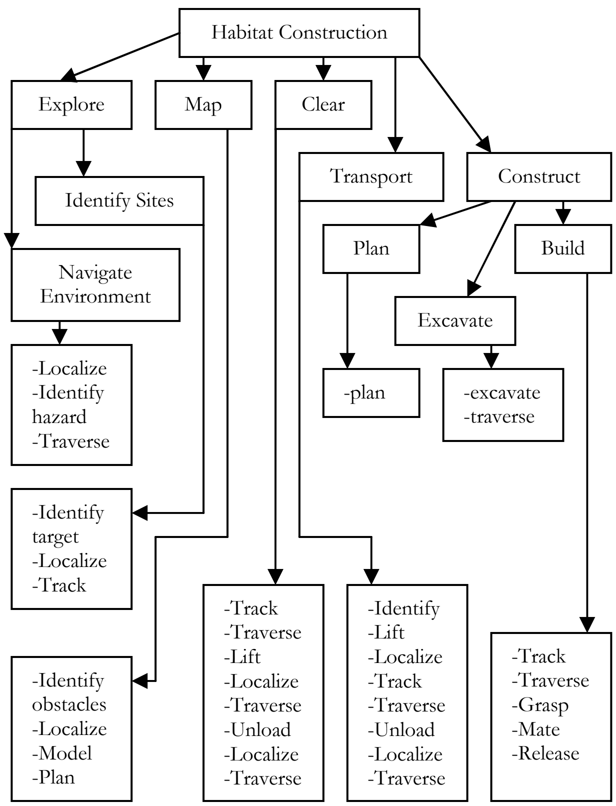

Figure 1 : Scenario Decomposition Scheme

We now describe in detail how each step is simulated in the Multi-Agent Robot Simulator for Lunar Construction (MARS-LC) system**.**

1. Explore

Robots navigate the environment, and then visually identify prospective sites, finally selecting one. We break up the environment navigation into traversals of distance $d_{explore}$. There are $n_{explore1}$ such tasks. Similarly, there are $n_{explore2}$ such tasks of site identification. The *explore*$_2$ jobs can only be started when all $n_{explore1}$ jobs are completed. The value of $n_{explore1}$ is revealed as the exploration proceeds (online assignment) subject to a predefined maximum value due to power consumption constraints after which a decision has to be made.

2. **Map**

Accurate modelling of the site and planning the site preparation is done in this step. This can only begin when all the exploration tasks are complete. There are n_{map} such tasks that can be allocated.

3. **Clear**

The chosen site is cleared of all obstacles. We assume there are n_{clear} number of obstacles of maximum mass m_{clear} that have to be moved a maximum distance of d_{clear}. Assuming that the maximum mass is within the lifting capability of any assigned robots, it would appear that there are n_{clear} number of tasks. But this could actually be a complex task if a robot does not have the energy to transport the obstacle the entire d_{clear} but requires the task to be assigned to another robot for the remaining distance.

4. **Transport**

Construction materials are transported from the landing area to the construction site. The number of habitats, $habitat_n$, and the number of component modules and equipment per habitat, $habitat_p$, each of maximum mass $m_{transport}$, are known *a-priori*. The transport distance $d_{transport}$ is known at the start of the transport operation. The number of total transport tasks is $n_{transport} = habitat_n * habitat_p$; but these can each be decomposed into 2 or more tasks by breaking up the distance $d_{transport}$, as in step 3.

5. **Construct**

This is broken up into three subtasks :

a. **Plan** : Plan the construction. There are n_{plan} such jobs.
b. **Excavate** : Excavate the site. There are $n_{excavate}$ such task, each task excavating a volume, $vol_{excavate}$, of regolith and transporting it $d_{excavate}$ distance for disposal. A excavation overhead energy of $\varepsilon_{excavate}$ is required for every excavate task, due to the nature of the lunar environment.
c. **Build** : Set up the inflatable components, attach them together, inflate and pressurize. There are n_{build} such tasks, all known *a-priori*.

The operations described above have certain sequential constraints, while some operations can be done in parallel. The partial ordering of the operations is as follows : Explore-Navigate $\prec$ Explore-Identify $\prec$ Map $\prec$ Clear $\prec$ Transport $\prec$ Construct-Build ; Map $\prec$ Construct-Plan $\prec$ Construct-Excavate $\prec$ Construct-Build.

Basic physics, using the parameters of the functional primitives, is used to model the energy and time consumption of the various tasks. We assume robots are of uniform mass m_{robot} and travel at average velocity v_{robot}. The lift distance of a manipulator arm is d_{lift}. Lunar gravity is *gravity* and regolith density is *density*. Robots fully charged, have a maximum energy reserve of ε_{robot}. We also assume that masses of all objects are within the lift capacity of a single robot.

The uncertainty in the problem domain is introduced by the fact that the number of exploration and excavation tasks and number of obstacles to be moved, the distances for equipment to be transported, and obstacle and excavation waste disposal are unknown until just before execution of the associated tasks. The drain on energy thus can only be assessed just before the task so this is an estimation due to the failure rates associated with the functional primitives. We use a recharging primitive that requires a certain amount of energy to access, but fully recharges a robot with the incurrence of a significant time penalty.

4.3 Task Allocation Algorithm

The factors motivating our methodology consist of two overarching concepts. The first concept deals with the development of a high-level plan that decomposes a scenario into functional steps. The second concept deals with allocating tasks for each step, subject to task dependencies, resource constraints, and robot capabilities. Our approach differs from others in that we use the lunar construction scenario to demonstrate an algorithm that handles different stages of task generation and allocation. We use the basic MURDOCH [6] market-based auction algorithm as our foundation but we extend it to add stages and complex task decomposition.

The lunar simulation problem, as described, has a distinct set of stages. The first three stages of *Explore-Navigate*, *Explore-Identify* and *Map* can be executed as three distinct iterations. In the two exploration stages, navigation tasks are introduced in an online manner and then suitable sites are identified. Mapping of the selected site is done in the *Map* stage. Following these stages is a composite stage that involves the operations of clear, construct-plan, construct-excavate and transport. All of these tasks are revealed before the start of this stage, but the tasks have sequential constraints amongst them. In addition, some tasks may be complex tasks. The constraints and complex tasks have to be handled here. The final stage is the construct-build stage, where all the tasks are known *a-priori* and can be assigned in one step or iteratively. Thus, the lunar simulation problem requires the use of a composite task allocation methodology to solve the task allocation problem. Our approach uses such a composite approach. While we do use domain knowledge of the tasks and their order and mode of arrival into the system to improve our performance, we attempt to generalize the broad principles of our algorithm.

Algorithm Outline

1. Devise a broad plan of the task scenario, breaking it into stages
2. Define highly specific metrics for each complex task in a scenario using domain knowledge
3. Process each stage of the problem. For each stage
   a. Obtain next task for announcement subject to any sequence constraints or online discovery

b. Designate an auctioneer from among the robots. Auctioneer announces next task to idle robots based on functional primitives of task with {name, energy requirements, metrics} and then waits for candidates to send their bids computed based on the metrics for the task
c. Auction closes, losers notified, winner given a contract
d. Auctioneer monitors the task. If insufficient progress is being achieved, the contract is canceled, the allocated task is retracted and reintroduced for re-bidding
e. If an allocated task is a complex task,
   i. Winner assesses its ability to complete task based on energy requirements
   ii. If it cannot complete the entire task, winner attempts to partially complete task. The uncompleted portion is then sent back to the auctioneer to be reintroduced for bidding.

5 Evaluation

5.1 Metrics

The performance of any market-based system will depend on the metrics used to compute the utility values. We experimented with different methods and tried to define a metric that would capture the important variables of our problem. Our metric is composed of three components:

i. **Minimal Matching Score**: We define this to be the difference, in number of skills, between the skill set of a robot and the required skill sets of the task under consideration. This component is minimized to reduce the number of over-qualified robots assigned to tasks. The intuitive idea is the better a robot is suited for a job, the better the overall task allocation.
ii. **Estimated Energy and Time**: Each robot estimates the amount of time and energy it would need to complete the task.
iii. **Available Energy**: Greater preference is given to those robots that would be able to complete the task without a recharge.

These 3 components are normalized to a value between 0 and 1. We then compute the weighted sum of these components as our overall utility. The weights are computed using the *Analytic Hierarchy Process* [19]. The input to this process is a matrix of pair-wise comparisons between each of the components. The value of each comparison, ranging from 1-9, defines how strongly a component is preferred over the other component, with 1 being no preference and 9 being absolutely preferred. Our primary preferences were first to avoid recharges and then to minimize estimated time.

5.2 Experimental Setup

The Multi-Agent Robot Simulator for lunar Construction (MARS-LC) is built on the Teambots environment [20] and was used to test our methodology in various scenarios.

Three different strategies for task allocation were considered. In the *Non-preemptive* approach, once a robot was assigned a task, the assignment could not be retracted. If the robot did not have enough energy to complete the task, it would have to recharge, and this recharge time penalty would have to be borne. In the *Preemptive* approach, robots that were recharging could be preempted from their assigned tasks and these tasks could be re-auctioned. In the *Complex Task* approach, in addition to having preemption, *clear* and *transport* tasks did not have to be integral but could be decomposed into sub tasks and performed in chunks.

We ran experiments on an area of 100 x 100 meters. The base camp was in the center of this grid, and we choose camp and disposal site locations with typical parameters. An average of 8 robots was used in each set of experiments, with each set consisting of 100 trials. We varied recharge time in each trial and observed the total time taken to complete the task scenario. We also experimented with heterogeneous (4-2-2 breakup) vs. homogeneous robots and using metrics vs. a first come first served approach.

5.3 Discussion

As expected, the Preemptive method outperforms the Non-Preemptive method. The performance improves by an average of 20% as seen in Figure 2. The Complex Task allocation performs only as well as the Preemptive approach. This is probably because in the environment settings we used, most tasks could not be broken down into subtasks well enough to observe a performance difference. We believe that as the size of traversals becomes larger, the Complex Task method will outperform the other two approaches. In addition, it will also allow robots to do jobs that were not possible in the other two approaches, as when transporting a heavy object a large distance requires more energy than the energy of a fully recharged robot.

In our experiments with metrics, in the Preemptive approach, the addition of metrics significantly improves performance for homogeneous robots, as seen in Figure 3. Surprisingly, metrics make no difference in the performance of heterogeneous robots, and heterogeneous robots are outperformed by homogeneous robots. This could be explained by the fact that specialized skills automatically constrain task allocation in a particular direction, and the utility values are dominated by the overriding skill requirements of the tasks and the actual presence of those skill sets in the available robots.

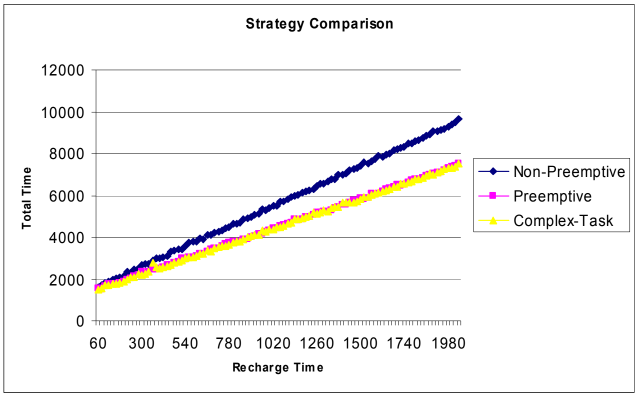

Figure 2. Experimental Results – Strategy Comparison

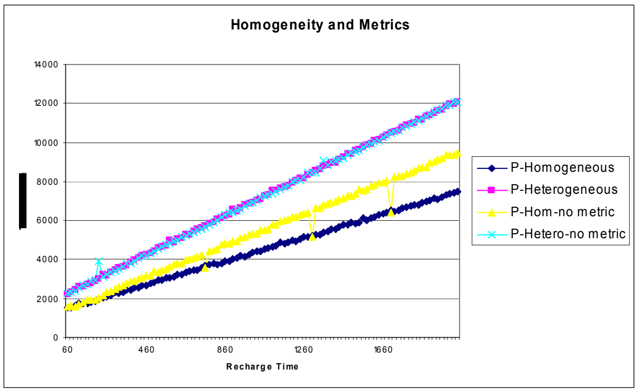

Figure 3. Experimental Results – Homogeneity vs. Metrics

6 Conclusions

In this paper, we have implemented a simulation of a lunar construction mission and shown how task allocation can be done in the different stages, among the various tasks of the mission. We have presented the results of our allocation strategy with non-preemptive, preemptive, and complex task allocation, and shown how we can improve the task reallocation. In the future, we will conduct more extensive testing with complex task allocation and more finely tune our metrics to the specifics of each task. We also plan to expand our simulator, bringing it closer to the real world problem and to use a world model of the domain to add learning and reasoning to better allocate the tasks.

References

[1] NASA, "The Human and Robotics Technology Formulation,"http://exploration.nasa.gov/documents/documents.html, 2004.

[2] Mataric M., Sukhatme, G., Ostergaard E, Multi-Robot Task Allocation in Uncertain Environments, Autonomous Robots 14, 244-263, 2003.

[3] Gerkey, B., Mataric, M, A formal analysis and taxonomy of task allocation in multi-robot systems, Intl. J. of Robotics research 23(9): 939-954, September 2004.

[4] Becker R., et. al., "Transition-Independent Decentralized Markov Decision Processes," Proc. 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems, July 2003.

[5] Botelho S, and Alami R, "M+: A Scheme for Multi-Robot Cooperation Through Negotiated Task Allocation and Achievement," ICRA 1999.

[6] Gerkey, BP and Mataric, MJ, "Sold!: Auction methods for multi-robot coordination," IEEE Transactions on Robotics and Automation 18(5), 758-768, 2002.

[7] Chaimowicz, L., Campos, M., Kumar, V. Dynamic Role Assignment for Cooperative Robots, ICRA 2002, Washington, D.C., May 11 - 15, 2002.

[8] Parker L., "ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation," *IEEE Transactions on Robotics and Automation*, 14 (2), 1998.

[9] Werger B.B. and Mataric M., "Broadcast of Local Eligibility: Behavior-Based Control for Strongly-Cooperative Robot Teams," *Proceedings of Autonomous Agents 2000.*

[10] Zlot, R.M. and Stentz A., "Complex Task Allocation For Multiple Robots," *Proc. Int. Conference on Robotics and Automation*, April, 2005.

[11] Benaroya H., Bernold L. et al, Engineering, Design and Construction of Lunar Bases, Journal of Aerospace Engineering, April 2002

[12] Howard A., and Rodriguez G., "Validating Mission Relevance of Autonomy Technologies through Increased Science Return," 20th International Conference on Machine Learning, pgs. 31-35, August 2003.

[13] Rodriguez G. and Weisbin, C. "A New Method to Evaluate Human-Robot System Performance," Autonomous Robot, 14(3), March 2003.

[14] Huntsberger T.L., Rodriguez G., and Schenker P.S., "Robotics challenges for robotic and human Mars exploration," *Proc. ROBOTICS2000,* Albuquerque, NM, Mar 2000, pp. 340-346.

[15] Schenker P., et al, Robot work crews for planetary outposts: close cooperation and coordination of multiple mobile robots, 2000.

[16] Pirjanian P. et al, "Development of CAMPOUT and its further applications to planetary rover operations" SPIE Symp. on Sensor Fusion and Decentralized Control in Robotic Systems III, Boston, MA, Nov. 2001.

[17] Lincoln W.P, et. al., "Relative Benefits of Potential Autonomy Technology Investments," *Proc. Int. Conf. on Space Mission Challenges for Information Technology,* Pasadena, CA, 2003.

[18] Howard A., et. al. "A Methodology to Determine Impact of Autonomy Technologies on Space Science Mission," Int. Symp. Robotics and Applications, Spain, 2004.

[19] Saaty T.L., The Analytic Hierarchy Process, NY, McGraw Hill, 1980.

[20] Teambots, http://www.teambots.org